Towards Trustworthy Shared Sensor-Actuator Networks

[Extended Abstract]

Ramy Eltarras
Electrical and Computer Engineering, Virginia Tech
ramy@vt.edu

Mohamed Eltoweissy
Pacific Northwest National Lab
mohamed.eltoweissy@pnl.gov

Stephan Olariu
Computer Science, Old Dominion University
olariu@cs.odu.edu

Ing-Ray Chen
Computer Science, Virginia Tech
irchen@cs.vt.edu

1. INTRODUCTION

We are witnessing a rapid expansion in the adoption of networked sensor-actuator systems (NSAS) deployed in support of applications such as smart homes, health management, public safety, and emergency management. Many of these emerging applications require large-scale deployment of NSAS and often have dynamic application-specific mission and evolving quality-of-service (QoS) requirements that include timeliness, reliability, security and availability. The shared and federated use of NSAS resources, to achieve multi-application goals, is a key to cost effective NSAS industry. This necessitates the decoupling of the NSAS physical infrastructure from application provisioning, and protecting applications and infrastructure resources from threats. The failure of NSAS nodes, due to malicious or non-malicious conditions, represents a major threat to the trustworthiness of NSAS platforms. Applications should be able to survive individual failures of resource nodes and change their runtime structure while preserving its operational integrity. Furthermore, for sustainable operation, QoS provisioning must be interwoven with energy conservation as a core priority in NSAS platform design. The large-scale of such networks, their heterogeneous node capabilities, their highly dynamic topology, their resource challenged nodes with the subsequent need for node cooperation, and their likelihood of being deployed in inhospitable environments, pose formidable challenges for the construction of trustworthy, shared NSAS platforms. To support the transition of NSAS from a research-only topic to a cost-efficient commercial industry that brings NSAS products and technologies to market, there is a need for a system-aided engineering methodologies and processes that addresses the industrial activities required for the full life-cycle of NSAS applications starting from the initial design to the evolution as requirements or mission change.

In this paper, we overview our ElaStic Shared nEtworked seNsor-aCtuator systEm (ESSENCE) project that aims to realize a scalable, trustworthy, shared NSAS platform endowed with adaptive configuration, networking and QoS management capabilities. The technical merit and novelty of this work lie in laying the foundation for a systematic, disciplined, quantifiable approach to the design, provisioning, maintenance, and evolution of this class of systems incorporating *role*, *route* and *redundancy* engineering as effective tools to best satisfy the varying needs of NSAS applications. To the best of our knowledge, there are no widely accepted design principles or engineering methodology that addresses the full lifecycle of such adaptive shared NSAS platforms.

This work is sponsored in part by NSF award 0721523

The problem of dynamic self-configuration in NSAS has recently attracted the attention of researchers. Research efforts have addressed the problem from two different perspectives: (1) Dynamic topology reconfiguration; and (2) Dynamic software reconfiguration. Our focus is on the latter, more recent, category. While there are many active research efforts addressing NSAS programming and reconfiguration, most of the proposed models and software architectures suffer from serious limitations. Federated and shared use of sensor networks was addressed in [3]. In [6] an autonomous NSAS platform was proposed to enable secure dynamic task-based networking and in-network storage. A role-based approach for the hierarchical self-organization of sensor nodes was presented in [5].

2. OVERVIEW OF ESSENCE

ESSENCE attempts to realize a scalable, trustworthy, flexible, cost-effective and dynamic computing and communication infrastructure capable of efficiently running multiple applications on top of shared NSAS resources. ESSENCE is based on three cornerstones, namely role, route and redundancy engineering. The outcome is that an application using ESSENCE would be allocated, at runtime, a dynamic taskforce, running over the ESENCE resource pool that would satisfy its evolving mission requirements. The application is oblivious of the ESSENCE resource management while ESSENCE exploits application semantics for efficiency. ESSENCE perceives both applications and the NSAS itself to be elastic, and allows them to grow or shrink based on needs and conditions. To realize this vision, we propose a truly general purpose runtime platform designed specifically to support the engineering practices required to build and manage the full lifecycle of NSAS applications.

ESSENCE enables QoS-sensitive, long-lived applications and services to be deployed over trustworthy, survivable NSAS platforms and promotes wider adoption of trustworthy shared NSAS technology in our daily lives enhancing (micro-level) interaction with our physical surroundings and enabling unprecedented breakthroughs in critical areas like healthcare,

sustainable environments, transportation, emergency response and public safety. ESSENCE contributions belongs to two categories: technical, and methodological. Our general approach is to support dynamic node re-tasking and network reconfiguration, efficient addressing and routing over large-scale NSAS with anonymous nodes, redundancy management, and dynamic QoS provisioning and resource allocation. Specifically, we (1) develop the ESSENCE architecture to define and manage dynamic role assignment for nodes, to enable NSAS to dynamically change and reconfigure the allocation of their resources to applications according to their evolving mission needs; (2) develop generic addressing and routing that seamlessly supports different communication patterns (unicast, multicast, broadcast, and anycast) while, at the same time, exploiting application semantics for efficiency; and (3) develop a QoS provisioning scheme that recruits the workforce for an application by exploiting inherent redundancies and heterogeneity in NSAS while maximizing network lifetime from the application's perspective.

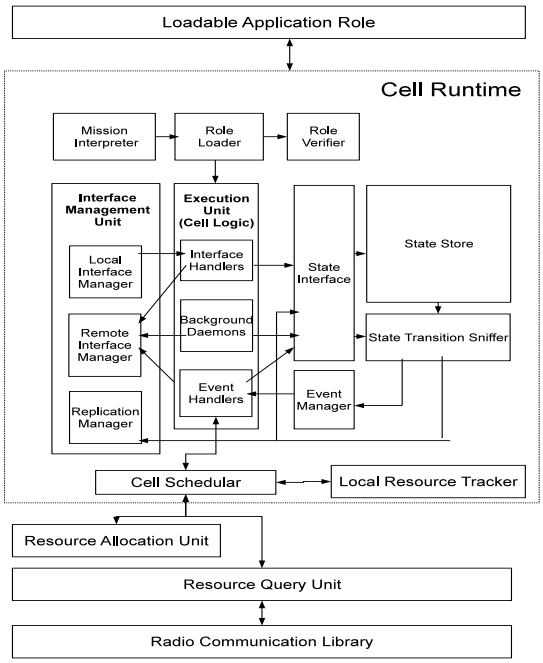

Figure 1: Cell Runtime Architecture

Role engineering involves enabling node and network reconfiguration by the dynamic assignment and deployment of roles in support of dynamic missions. By downloading a specific role(s), a sensor node essentially acts as a virtual machine in support of a given mission. Reconfiguration should also be enabled in-network whereby a node can discover and pull role code from other nodes. Our proposed research aims to provide both capabilities. Such reconfiguration requires managed update of the system software as well as a highly dynamic runtime platform capable of executing the updates. We propose a new architecture called Cell-Architecture (CA) to support the execution of role oriented NSAS software. The Cell is a generic virtual computational unit that realizes multiple role-players per physical sensor node. The software designer views the NSAS system as a set of interacting roles executed ubiquitously on top of a set of role-playing cells. Inspired by the biological stem cells, the CA cells have the ability to specialize into a specific organ cell performing a specific role. Unlike stem cells, Essence cells can change their roles after initial specialization. Accordingly, CA achieves a high level of decoupling between roles and cells by allowing the reassignment of roles at runtime and by providing a convenient and efficient mechanism to dynamically reconfigure NSAS nodes. Figure 1 shows the architecture of ESSENCE cell runtime.

Route engineering involves providing addressing and routing capabilities that seamlessly support various communication patterns (from unicast to anycast) and also exploit the application semantics to adapt route construction in order to support the efficient operation of long-lived shared NSAS platforms. Although many routing protocols have been proposed for wireless sensor networks, most of the proposed protocols either do not exploit query semantics or are designed with a specific application or load pattern in mind. A truly generic, flexible, and adaptive routing solution that can exploit query semantics is essential for the efficient operation of such large-scale shared NSAS platforms. We propose Adaptive Multi-Criteria Routing (AMCR) that provides generic criteria-based addressing of resources and allows each application to publish its own resource criteria. AMCR exploits application-specific query semantics by allowing applications to create routing indexes for the most frequently queried criteria. AMCR also supports multi-criteria indexes to efficiently route complex queries. AMCR is the communication backbone of ESSENCE.

Redundancy engineering involves utilizing cross-layer information and dynamically identifying optimal settings along different dimensions of redundancy such as path, source, and modality to not only satisfy QoS requirements of concurrent applications, but also to promote the survivability of the system under varying conditions. While uncontrolled redundancy may help satisfy availability, reliability and timeliness, it does not scale well and invariably leads to excessive energy consumption. We devise quantitative redundancy parameters and develop a formal model to explore tuning of redundancy parameter values to satisfy QoS and energy requirements subject to varying network conditions.

2.1 Role Engineering

The role is a commonly used design concept that facilitates the separation between roles from role players for both system design and permission management. The role oriented adaptive design (ROAD) was introduced in [1]. ROAD provides a framework for building adaptive general purpose software system, however the principle fits well NSAS software. Role-Based Access Control (RBAC) is a standardized access control mechanism commonly used in enterprise systems[2]. Role engineering refers the engineering methodology for designing, implementing, deploying, securing, managing, and evolving role oriented software systems. We employ role engineering to provide a general platform enabling dynamic self reconfiguration of shared NSAS. The application provides the implementations of roles and the network provides the role players. The assignment/reassignment of roles to players is dynamic and maintained by the runtime environment. We also employ role-based access control to

improve security and accountability. We achieve dynamic reconfiguration in ESSENCE using role engineering that results in software systems that separates roles from their players. Such separation allows transparent reassignenment of roles among independently managed role players. It also supports the structural plasticity of the software facilitating evolution. By adopting role oriented design in NSAS, the functional and organizational structure of software can be sustained despite the continous changes in the underlying physical resources of the network. Decoupling the application roles from the role players, allows the application designers to focus on the application logic rather than the underlying network resources, and enables the dynamic reconfiguration of the network when needed to achieve fault tolerance, longer lifetime, and meeting specific QoS requirements. We propose a novel middleware architecture to support the execution of role oriented software in NSAS environment and enable dynamic reconfiguration through managed update of the system software. To our knowledge, no previous work attempted to realize runtime environment for role oriented adaptive software in NSAS.

2.2 Route Engineering

We propose the Adaptive Multi-Criteria Routing (AMCR) framework and protocol suite as the first NSAS routing approach to provide the following three interrelated contributions: (1) highly Expressive Addressing Scheme, (2) seamless unicast,multicast,anycast, and broadcast, and (3) exploiting message semantics and node capabilities. AMCR greatly enhances the trustworthiness of the network by supporting role and path redundancies. AMCR provides a sharp departure from most current addressing schemes that utilize a unique identifier for each network node, which limits its application scalability, reliability and imposes significant overhead for resource and identifier management. Instead, AMCR adopts a criteria-based addressing scheme. AMCR allows destination addresses to be specified as a descriptive reference to node capabilities, administrative settings, and/or application published criteria. In AMCR addresses are predicates referencing nodes. As it turns out, other addressing schemes, including IP, are subsumed by AMCR and can readily be realized using the AMCR addressing scheme. This enables AMCR to seamlessly route unicast, multicast, anycast, and broadcast traffic using the same addressing and routing technique so that route information for one type of routing increases routing knowledge of other types. Applications only specify a descriptive destination predicate that may match one, some, or all nodes. Application can also limit the number of destination nodes achieving anycast communication capability.

Most of the routing approaches for NSAS attempt to exploit message semantics for better efficiency at the cost of being application-specific. AMCR attempts to optimize communication efficiency through a novel criteria-indexing mechanism that adapts to the observed application semantics and load patterns without sacrificing its generic properties. AMCR is designed as a general-purpose routing protocol that does not restrict the application's choice of desired communication patterns, data reporting model, or mobility model. A novelty of the AMCR design is that it provides a truly generic routing protocol capable of exploiting the message semantics and adapting itself to the observed application characteristics in order to support the efficient operation of trustworthy shared NSAS platforms.

AMCR adopts a targeted messaging model that allows the exchange of messages between a source node and a targeted group of nodes. The source node specifies its targeting criteria expressed through a Boolean expression named *target predicate*. A *targeted message* is a network packet carrying application data in its payload and *target predicate* in its header. AMCR is responsible for routing *targeted messages* to their destination groups and optionally routing responses back to the source. This model is highly generic for NSAS as it bases its assumptions on the system level properties of NSAS environment rather than on any class of NSAS applications. AMCR recognizes the heterogeneity in node capabilities and maintains a profile for the NSAS node based on its capabilities, spatial location, and other administrative criteria. We refer to this profile as node profile. Similarly, each cell has its own cell profile that inherits all the criteria in its host's node profile and includes additional application-specific criteria. A profile is an aggregation of criteria. Each criterion is key/value pair. Application can construct target predicates based on criteria values. The main principle behind AMCR is to provide the flexibility of arbitrary predicates in queries and minimize the route discovery overhead for frequent queries in a way similar to creating indexes for the frequently queried database fields. The working set of indexes adapts to the observed workload; the index not used for a long time is not maintained, and new query patterns results in the creation of new index.

Applications can define their own criteria. This provides a convenient and efficient way for application to group resources and implement collaboration oriented tasks. Network management components can also use AMCR for resource management, fault tolerance, redundancy, optimization, and enforcing quality of service constraints. Unlike traditional IP-based networks, addressing in AMCR does not require a unique identifier for every single node. Instead, the address in AMCR is a highly descriptive predicate that can match one or more nodes depending on the criteria values of the nodes and conditions specified in the predicate. AMCR does not require all nodes to play a router role. For AMCR a node can be network resource, a router, or both.

2.3 Redundancy Engineering

We propose to satisfy application QoS requirements while maximizing NSAS longevity by determining the optimal level of multi-dimensional redundancy dynamically in response to network dynamics. Our focus is on reliability and timeliness as application QoS parameters and on controlling redundancy at the "source," "path" and "modality" levels to achieve the desired QoS requirements. The "source" level redundancy is the use of multiple sensors to return the requested sensor reading. The "path" level redundancy refers to the use of multiple paths to relay the reading to the sink node. The "modality" level refers to the use of sensors and actuators of distinct types (biometrics, acoustic, pressure, temperature, cell-phone-enabled, etc.) to report the same physical phenomenon such that there is enough information to infer the requested information even if not all sensors can successfully transmit their sensor readings to the sink. Since NSAS are resource constrained, the preliminary design

of ESSENCE utilizes hop-by-hop data delivery and dynamically forms multiple paths for data delivery without incurring extra overhead to first formulate multiple paths before data delivery. Redundancy engineering is part of ESSENCE and is primarily used to dynamically determine the best redundancy to satisfy the negotiated QoS requirements.

3. EVALUATION

We built a simulation framework for ESSENCE capable of capturing specific application behaviour. We defined a *network model* for ESSENCE based on *constrained connected random unit graphs* [7] and a *traffic model* based *idealized information traffic patterns*. The *traffic model* captures the run-time behaviour of NSAS applications according to an interactivity typology inspired by [4]. Complex traffic patterns are modelled as a stochastic random process generating traffic according to a weighted aggregation of multiple basic idealized traffic patterns. The simulation framework provides a tool for evaluating performance and efficiency gains of various classes of application with different routing needs. Due to space limitation, we only show results of our AMCR overhead study. Detailed experiments will be presented in sequel papers.

We prove that the index propagation overhead in AMCR equals to $(\frac{1}{h} - \frac{1}{h(p+1)})\lceil\frac{r-re^{\frac{-h}{\beta}}}{l}\rceil + \frac{k}{h(p+1)}\log\frac{n}{p+1}$ where , where h is the period of the index advertising cycle, p is the *router packing ratio*, n is the number of nodes in the network, r is the number of indexed criteria, β is the scale parameter of exponential probability distribution function representing the index update events, and l is the maximum number of index update records that fit in a single advertisement packet.

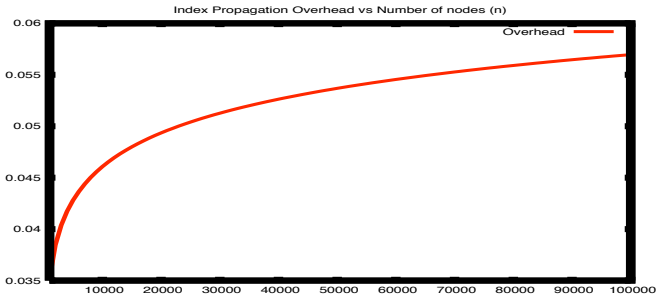

Figure 2: Index Propagation Overhead vs Number of nodes (n)

The scalability of index propagation is clearly shown in figure 2 where overhead is only slightly increased from 1000 nodes to 100,000 nodes. The index propagation overhead also increases in steps with the increase in the number of indexed criteria as shown in 3. The step effect is attributed to packetization.

4. CONCLUSION

We proposed ESSENCE, a novel platform for scalable trustworthy shared NSAS. ESSENCE addresses the full life-cycle of NSAS applications and divides the solution into three inter-related engineering concerns namely role, route, and redundancy to enhance the engineering process efficiency.

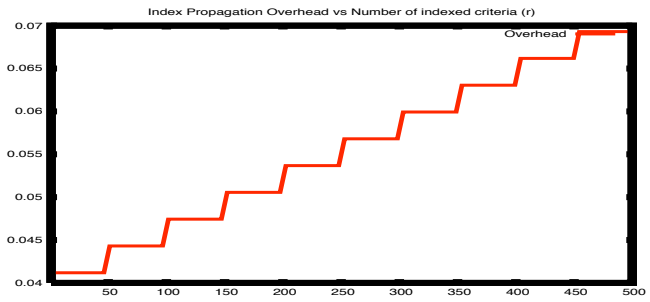

Figure 3: Index Propagation Overhead vs Number of indexed criteria (r)

We have shown that dynamically managing roles on NSAS resources, generically and seamlessly supporting varied communication patterns while exploiting application semantics in routing, and providing optimal per query redundancy (1) lead to efficient sharing of NSAS resources by dynamic long-lived applications while satisfying their QoS requirements, (2) maximize the useful lifetime of networked sensor-actuator systems, and (3) support changes in application and system requirements. We plan to investigate scalable performance isolation techniques for ESSENCE as well as domain-based data protection techniques. We also plan to extend ESSENCE to support the federation of resources owned by multiple entities and enabling logical mobility of NSAS services.

5. REFERENCES

[1] A. Colman and J. Han. Using role-based coordination to achieve software adaptability. *Sci. Comput. Program.*, 64(2):223–245, 2007.

[2] D. F. Ferraiolo, R. Sandhu, S. Gavrila, D. R. Kuhn, and R. Chandramouli. Proposed nist standard for role-based access control. *ACM Trans. Inf. Syst. Secur.*, 4(3):224–274, 2001.

[3] C. Huygens and W. Joosen. Federated and shared use of sensor networks through security middleware. *Information Technology: New Generations, Third International Conference on*, 0:1005–1011, 2009.

[4] J. F. Jensen. The concept of interactivity – revisited: four new typologies for a new media landscape. In *UXTV '08: Proceeding of the 1st international conference on Designing interactive user experiences for TV and video*, pages 129–132, New York, NY, USA, 2008. ACM.

[5] M. Kochhal, L. Schwiebert, and S. Gupta. Role-based hierarchical self organization for wireless ad hos sensor networks. In *WSNA'03*, pages 98–108. ACM, 2003.

[6] S. Olariu, M. Eltoweissy, and M. Younis. Answer: autonomous wireless sensor network. In *Q2SWinet '05: Proceedings of the 1st ACM international workshop on Quality of service & security in wireless and mobile networks*, pages 88–95, New York, NY, USA, 2005. ACM Press.

[7] F. A. Onat, I. Stojmenovic, and H. Yanikomeroglu. Generating random graphs for the simulation of wireless ad hoc, actuator, sensor, and internet networks. *Pervasive Mob. Comput.*, 4(5):597–615, 2008.